Review of Agent Models

Encounters, Strategies, Learning and Evolutions

Luis Mateus Rocha
Complex Systems Modeling Team
Computer Research and Applications Group (CIC-3)
Los Alamos National Laboratory
Los Alamos, NM 87545
e-mail: rocha@lanl.gov or rocha@santafe.edu
WWW: http://www.c3.lanl.gov/~rocha

1. AGENTS

The term agent is used today to mean anything between a mere subroutine to a conscious entity. There are "helper" agents for web retrieval and computer maintenance, robotic agents to venture into inhospitable environments, agents in an economy, etc. Intuitively, for an object to be referred to as an agent it must possess some degree of autonomy, that is, it must be in some sense distinguishable from its environment by some kind of spatial, temporal, or functional boundary. It must possess some kind of identity to be identifiable in its environment. To make the definition of agent useful, we often further require that agents must have some autonomy of action, that they can engage in tasks in an environment without direct external control. This leads us to an important definition of an agent from the XIII century, due to Thomas Aquinas: an entity capable of election, or choice.

This is a very important definition indeed; for an entity to be referred to as an agent, it must be able to step out of the dynamics of an environment, and make a decision about what action to take next – a decision that may even go against the natural course of its environment. Since choice is a term loaded with many connotations from theology, philosophy, cognitive science, and so forth, I prefer to discuss instead the ability of some agents to step out of the dynamics of its interaction with an environment and explore different behavior alternatives. In physics we refer to such a process as *dynamical incoherence* [Pattee, 1993]. In computer science, Von Neumann, based on the work of Turing on universal computing devices, referred to these systems as *memory-based systems*. That is, systems capable of engaging with their environments beyond concurrent state-determined interaction by using memory to store descriptions and representations of their environments. Such agents are dynamically incoherent in the sense that their next state or action is not solely dependent on the previous state, but also on some (random-access) stable memory that keeps the same value until it is accessed and does not change with the dynamics of the environment-agent interaction. In contrast, state-determined systems are dynamically coherent (or coupled) to their environments because they function by reaction to present input and state using some iterative mapping in a state space.

Let us then refer to the view of agency as a dynamically incoherent system-environment engagement or coupling as the *strong sense of agency*, and to the view of agency as some degree of identity and autonomy in dynamically coherent system-environment coupling as the *weak sense of agency*. The strong sense of agency is more precise because of its explicit requirement for memory and ability to effectively explore and select alternatives. Indeed, the weak sense of agency is much more subjective since the definition of autonomy, a boundary, or identity (in a loop) are largely arbitrary in dynamically coherent couplings. Since we are interested in simulations of decision-making agents, we need to look in more detail to agent based models with increasing levels of dynamical incoherency with their environments.

To summarize:

1. ***Dynamically Coherent Agents***
   a. Rely on subjective (spacial, functional, temporal) definition of autonomy. Function by reaction and are dynamically coupled to environments.
   b. Example: situated robots (wall-following machines), state-determined automata.
2. ***Dynamically Incoherent Agents***
   a. Possess models, syntax, language, decision-making ability. In addition to a level of dynamical coherence with their environments (material coupling), they possess an element of dynamical incoherence implemented by stable memory banks.
   b. Example: anything with symbolic memories and codes.

2. AGENTS: HOLLAND'S COMPLEX ADAPTIVE SYSTEMS

John Holland [1995] defines agents as rule-based input-output elements whose rules can adapt to an environment. These rules define the behavior strategy utilized by agents to cope with a changing environment. He also defines 7 basics or characteristics of agents and multi-agent sysems which further specify the rule-based adaptive behavior of agents:

1. **Aggregation** (Property) has 2 senses
   a. Categorization (agent level): Agents cope with their environments by grouping things with shared characteristics and ignoring the differences.
   b. Emergence of large-scale behavior (multi0agent level): From the aggregation of the behavior of individual agents (e.g. Ants in an ant colony) we observe behavioral patterns of organization at the collective level. This leads to hierarchical organization.
2. **Tagging** (Mechanism). Agents need to be individualized. They possess some identity. This in turn facilitates selection, specialization of tasks, and cooperation as different specific roles and strategies may be defined.
3. **Nonlinearity** (Property). The integration or aggregation of agents in multi-agent systems is most often non-linear, in the sense that the resulting behavior cannot be linearly decomposed into the behavior of individual agents. This also implies that multi-agent systems lead to network causality, as effect and cause of agent behavior follow circular loops that cannot be linearly decomposed into traditional cause and effect chains.
4. **Flows** (Property). Multi-agent systems rely on many connections between agents that instantiate the flow and transfer of interactions, information, materials, etc. Typically, the network of flows is represented with graphs.
5. **Diversity** (Property). Typically, multi-agent systems are heterogeneous, as there exist different agent roles and behaviors. This makes the tagging mechanisms important so that these different roles and behaviors may be identified.

6. **Internal Models** (Mechanism) organize the rules that produce agent behavior and can be used to let agents anticipate expected inputs from the environment. We can divide models in 2 types.
   a. *Implicit*: Prescribes a current action under implicit prediction. This is associated with hard-wired rules of behavior (e.g. by natural selection) and implemented by state-determined automata. This kind of model instantiates agents which are dynamically coupled to their environments, e.g. reactive, situated robots.
   b. *Explicit*: Use Representations stored in stable (or random access) memory to look ahead by exploring possible alternatives. This type of model produces agents with a level of dynamical incoherence with their environments, since they act not only based on current state and input but also by integrating information stored in memory. This integration can be pursued with more or less complicated reasoning procedures. Since agents with explicit models possess behavior alternatives, we can use them to study decision processes.
7. **Building blocks** (Mechanism). Agents are built with less complicated components. This allows for the instantiation of coded construction, which is essential for the recombination of components to produce new agent with different behavior and models. Natural selection, for instance, acts on the ability to randomly vary descriptions of agents, which are cast on a language coding for building blocks leading to the production of new agents.

Holland's 7 basics lend themselves to our notions of dynamical incoherence, and therefore we can well use them to describe agents and (complex adaptive) multi-agent systems.

3. Review of Agent Models: From Strategies to Explicit Models and Beliefs

3.1 Encounters and Strategies

The agents in the models described in this section are based on game-theoretic strategies, using simple memory architectures. The environments in these models are defined exclusively by other agents, therefore there is really no level of dynamical coupling between agent and environment. Rather, these models aim to study only decision strategies and the evolution of strategies in an environment of other changing strategies. However, the strategies pursued by these agents rely on present state and a memory of only a small number of previous states and encounters. In addition, typically, the agent-rule updating is synchronous (all agents updated at the same time) and there is a determined behavior outcome. This results in dynamically coherent multi-agent systems, since agents cannot choose when and whether to participate, and their rules are determined by a short list of previous states.

3.1.1 Iterated Prisoner's Dilemma: Evolutionary Strategy Dynamics

Idealized model for real-world phenomena such as *arm-races* (Axelrod, 1984) and *Evolutionary Biology* (Maynard-Smith, 1982), iut of Game Theory as defined by Von Neumann, Economics, and Political Science. The dilemma is defined as follows: 2 individuals are arrested for committing a crime together are held in separate cells; no communication is allowed; both are offered the same deal to testify against the other (both know this); if one testifies (defects) gets a suspended sentence (S) and the other gets the total sentence (T); if both testify (defect), the testimony is discredited: both receive a heavy sentence (H); if neither one confesses (cooperate) both are convicted to a lesser sentence (L). These values must obey the following 2 conditions: (1) S>L>H>T and (2) 2L>S+T.

- Evolutionary Phenomena in Simple Dynamics, Kristian Lindgren [1991]. Finite memory strategies for the iterated prisoner's dilemma (IPD) were encoded in a genetic algorithm (GA). Each agent contains a strategy h_m, which specifies how the agent should play each iteration of the prisoner's dilemma game, based on a memory of size *m*. a_0 denotes what the opponent did in the last encounter, a_1 denotes what the agent did in the last encounter, a_2, what the opponent did in the one but the last encounter, and so forth. The *tit for tat*[1] (TFT) strategy is a strategy of memory *m*=0, as it only requires memory of what the opponent did last.
    - Details of the GA:
        - Encoding (see figure 1). After ordering the binary possibilities of the rule, only the right side of the rules is encoded as a genetic description. For

[1] Defect if opponent defected last, cooperate if opponent cooperated last. Start by cooperating.

TFT, the binary possibilities are: 0 or 1 (opponent defected or cooperated) and the rule specifies 0 or 1 respectively as a strategy. This wat, TFT is oencode as [01]. See figure 1 for an example of an extension of TFT for m=1.

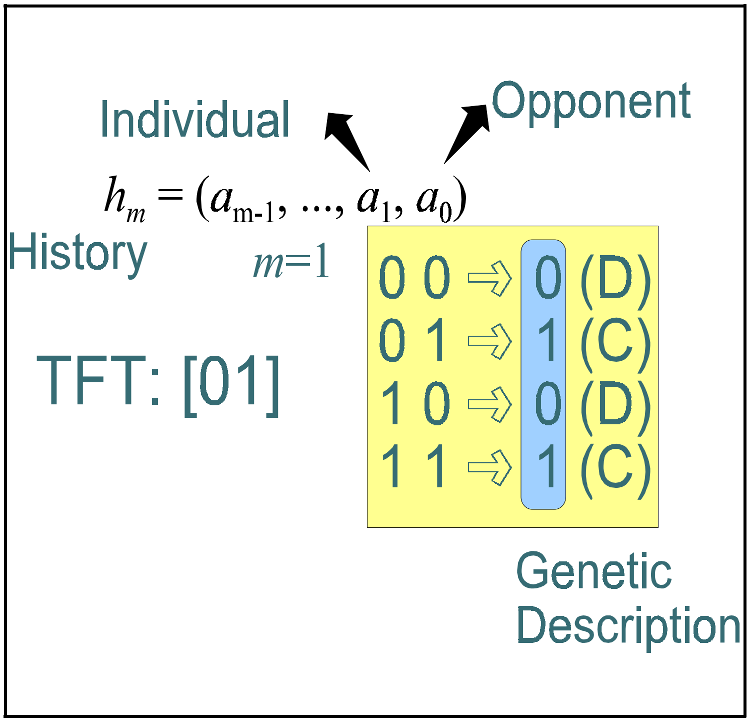

Figure 1: Encoding of IPD strategies in Lindgren's GA. C denotes cooperate, and D defect.

- The initial population of the GA possesses equal fractions of 4 strategies m=0: [00], [01], [10], and [11]
- Random operators. Point mutations, which flip a bit on the strategy: [01] ➜ [00] (2x10^{-5}); gene duplications which increase the memory size by 1 (do not change actual strategy): [01] ➜ [0101] (10^{-5}); and split mutations, which decrease the memory size by 1 producing two possibly distinct strategies: [1001] ➜ [10] or [01] (10^{-5}).
- Probability of mistake ***p***. Probability that rule will produce opposite outcome.

○ Lindgren's experiments show stasis, punctuated equilibria, varying speeds of evolution, mass extinctions, symbiosis, and complexity increase. Unlike Axelrod's and Rappoport's results previously, these experiments show that TFT is not evolutionarily stable as there are strategies which play as well with TFT but better against other strategies.

○ Experiments:
- Evolution of strategies of memory 1. See figure 2.

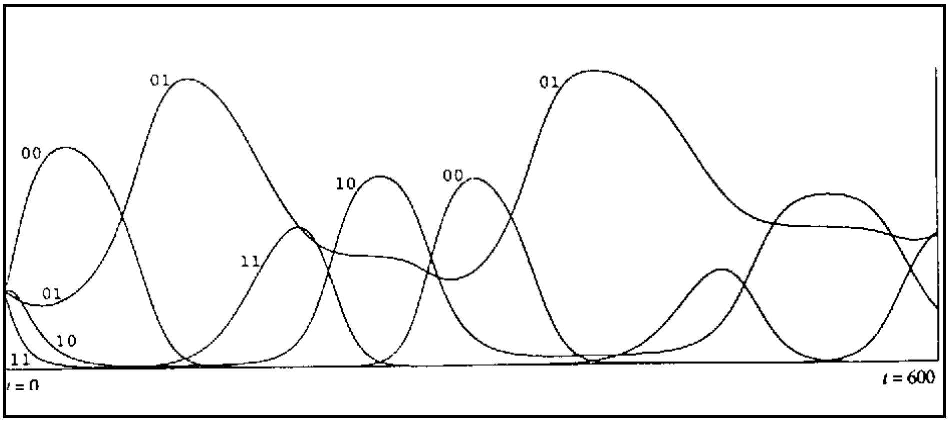

Figure 2: TFT [01] takes over "snakes" [00], the "suckers" [11] grow, and are exploited by Anti- TFT [10] and "snakes", but TFT eventually takes over again, etc. 600 generations.

- Evolution of strategies of memory 2 and 3. See figure 3.

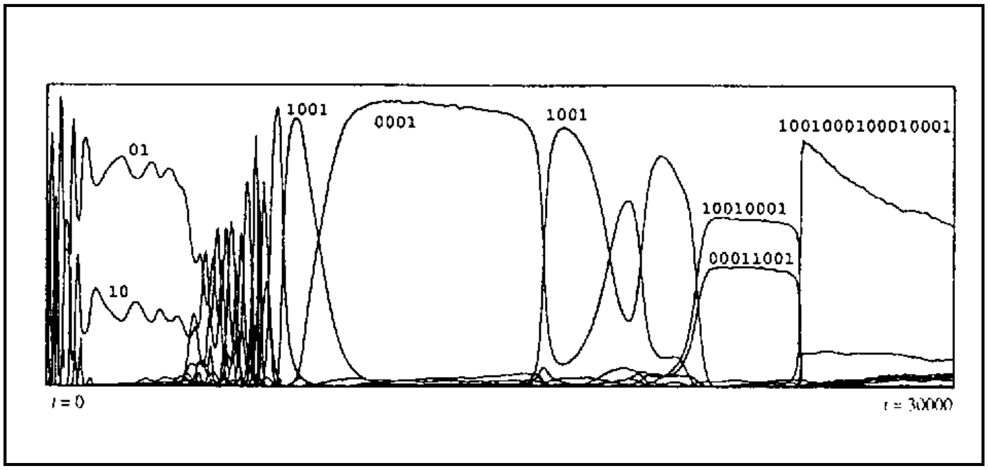

Figure 3: After a long period of strong oscillations we get to a long period of stasis with [0001], followed by oscillations with the arrival of memory 3 strategies in symbiosis (the success of [10010001] depends on the success of [00011001] and vice-versa). 30000 generations.

- Evolution of strategies of memory 4 and beyond. See figure 4.

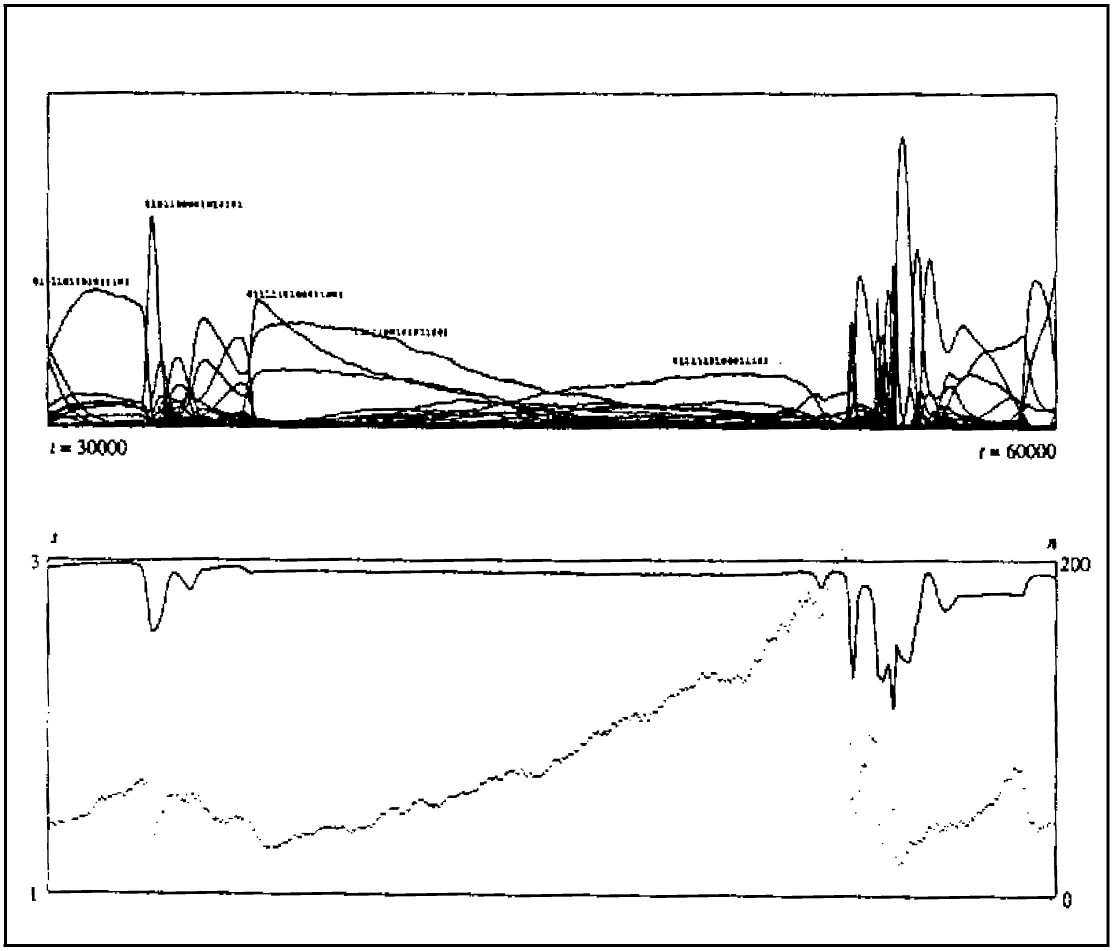

Figure 4: Memory 4 strategies appear establishing a class of 512 strategies that can lead [1xx10xxx0xxxx001]: Cooperative, but if one player accidently defects both players defect twice before returning to cooperative behavior. At this stage we also observe the phenomena of mass extinction. Bottom figure shows the average fitness (full) and number of different strategies (faint). Notice how we can expect periods of large diversity followed by radical mass extinctions. 600000 generations.

- OTHER RELATED WORK.
  - Karl Sigmund [1993] offers a very interesting overview of the original work by Axelrod and Rappoport on the IPD.
  - Melanie Mitchell [1996] and Klaus Emmeche [1994] offer an overview of Lindgren's experiments.
  - There is no evolutionary stable strategy in IPD [Boyd and Loberbaun,1987; Farrell and Ware, 1989]

3.1.2 Extending the Prisoner's Dilemma

- Including CHOICE AND REFUSAL OF GAME PLAYERS increases the emergence of cooperation and results in more interesting modeling of negotiation strategies as discovered by Stanley, Ashlock and Testfatsion, [1994]. This addition to the IPD tough not a tremendous computational extension, offers a qualitatively different way of modeling choice in the framework of the IPD. In this implementation, agents can refuse to play with other agents given a past history of encounters. This addition can be seen to increase the degree of dynamical incoherency between agents and their environment comprised of other agents as agents can opt to decouple themselves from the on-going strategy dynamics. Not surprisingly, this extension results in higher cooperation as agents can choose to play only with other cooperating agents, thus obtaining higher payoffs.
- Violating the second rule of the PD payoff scheme yields (as expected) strategies that take turns exploiting one another [Angeline, 1994]
- Cooperation also emerges in non-iterated PD in spatially arranged environments [Oliphant, 1994]. In this implementation, agents play the game with other agents in their neighborhood. This allows us to study the emergence of spatially distributed families of strategies.

3.2 Learning and Evolution

The agents in the models described in this section possess more interesting environments with changing or non-trivial demands. The objective of these models is to study how learning and knowledge can interact with evolution.

The Baldwin Effect (organic selection): If learning helps the survival of organisms (more plastic behavior) then this trait should be selected. If the environment is fixed, so that the best things to learn remain fixed, the learned knowledge may be eventually genetically encoded via natural selection. Example: animals capable of learning to avoid a new predator or other environmental danger, will survive long enough to allow genetic variation to eventually discover that avoiding the danger is useful trait to possess at a instinctual, genetically determined level. Waddington [1942] referred to this process as genetic assimilation.

- HINTON AND NOWLAN [1987] EXPERIMENTS: Agents were modeled by a very simple neural network and evolution was modeled by a Genetic Algorithm:
  - ○ 1000 encoded neural nets with 20 potential connections. A connection can be present (1 ⇨ 25%), absent (0 ⇨ 25%), and learnable (? ⇨ 50%).
  - ○ Crossover, no mutation.
  - ○ Fitness: only one layout of 0's and 1's can solve the environment's problem (all others have 0 fitness) 2^{20} possibilities.

    - If an agent has incorrect 1 or 0 it will never solve the problem, but those with ?'s have the capacity to learn the correct pattern in 1000 learning trials
    - fitness function: (1 + 19*n*/1000). tradeoff between efficiency and plasticity.
- "Learning" algorithm: Random guessing (network guesses at each trial the value of its connections). It is not a proper algorithm but it allows the modeling of the plastic nature of knowledge.
- Without learning the fitness landscape would be a single spike, with the plasticity of learning it becomes a smoother curve (figure 5).

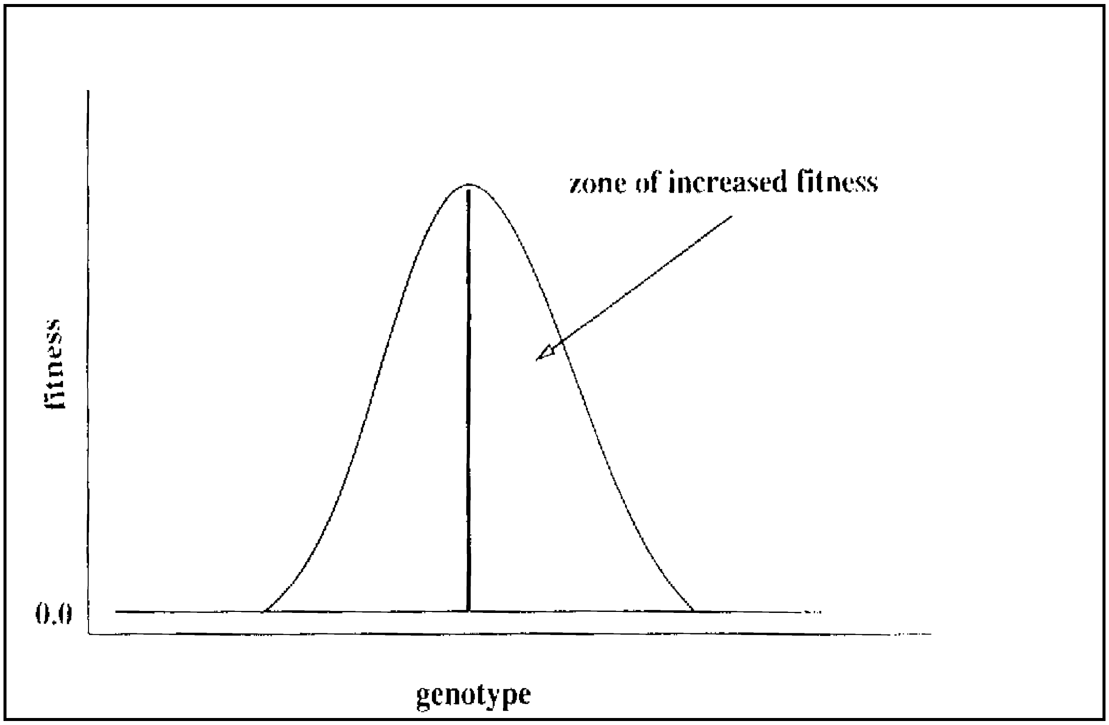

Figure 5: With the possibility of learning, the fitness landscape for the search problem is smoother, with a zone of increased fitness containing individuals able to learn the correct connection settings.

- Fitness never increases in runs without learning, but there is convergence to the solution when learning is introduced (figure 6, from [Belew, 1990]).

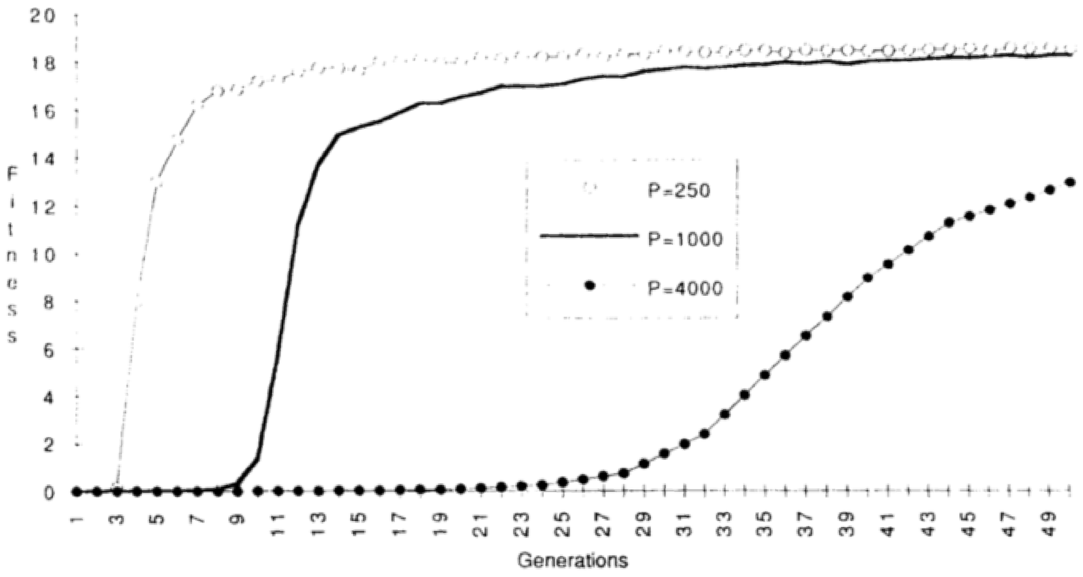

Figure 6: Fitness plot for three runs with 250, 1000, and 4000 agents in population.

- Learning can be a way for genetically coded partial solutions to get partial credit, allowing evolving agents to survive long enough for natural selection to discover the genetic solutions. Belew [1990] showed that after long runs, the number of unfixed positions (?) in the agents' networks decreases, as individuals with genetically encoded correct 1's and 0's eventually appear.
- These experiments show that even such trivial learning processes as random guessing in simple neural networks, will yield advantages to evolving agents in very harsh environments. Leaning provides ontogenetic plasticity to practically impossible phylogenetic solutions.

- Evolutionary Reinforcement Learning, [Ackley and Littman, 1991]. This model is a much more complicated exploration of the interactions between learning and evolution. It relies on an environment with no explicit evolutionary fitness function. Furthermore, unlike a genetic algorithm, agents in a population do not reproduce on every single generation, but only when certain necessary conditions are met. Agents are in this sense asynchronous. This imbues the model with a certain degree of dynamical incoherency between agents rules and environment laws which is both more realistic and important to model decision processes. Another very important aspect of the agent architecture in this model, is the distinction between innate, unchangeable, evaluation rules which denote the agents' "dispositions" or "beliefs" about their situation in the environment and the actual rules of behavior that can change in the lifetime if the agents. A similar architecture seems to be most appropriate to model decision processes for the agent models we envision for this project – even without an evolutionary component.

- The model:
  - Agents in a 2-D environment with food, predators, hiding places, etc.
- Agent architecture with 2 feedforward neural networks:
  - *Evaluation net*: gets agent state as input, produces judgement of how good state is. Fixed from birth (innate dispositions, goals and desires).
  - *Action net*: gets agent state as input, produces a move command (which can have many outcomes). Weights changed during lifetime with backpropagation/reinforcement learning algorithm. Implements the behavior of agents.
  - Fixed Architecture: only weights change for agents.

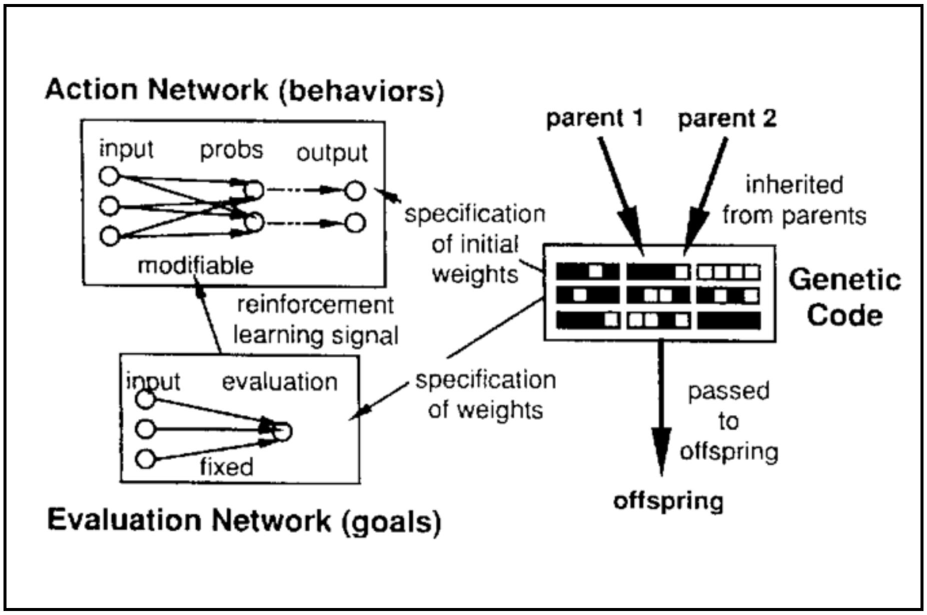

Figure 7: Agent architecture: action and evaluation networks. Architecture is encoded in the agents' genetic descriptions which are used, recombined, and mutated to produce new offspring.

- The evolutionary algorithm uses genetic descriptions for each agent. These descriptions encode (84) weights (of 4 bits) for networks (336 bits).
  - Agents have an energy gauge and die for low values. They must find food to keep energy at viable levels. Agents can only reproduce with enough energy.

    - Agents can reproduce asexually with mutation only and sexually when neighbor exists, with crossover and mutation.
    - Fitness emerges from environment and not from a pre- specified function (artificial life model).
- At each step of the computation the difference between current and previous evaluation is the reinforcement signal to modify weights of the action net. This way, agents learn to act in ways that lead to states “deemed” better by the agent’s own innate evaluation net.
- Agents then move based on the adjusted weights of action net.
- Performance: time to extinction of a population.
- ***Results***: Evolution alone (E) was not much better than random motion (B). Learning plus evolution (ERL) was not much better than learning alone (L). The authors propose that it is easier to generate randomly a good evaluation function than a good action function, since the latter network is simpler. This way, in runs without learning, the evolutionary algorithm cannot find a good behavior network. This is similar to the previous model: the environment is too harsh for evolution. On L runs, conversely, a good evaluation net is usually generated at random initially, and learning is then able to produce a good action bet, while evolution alone can not produce a good action net.

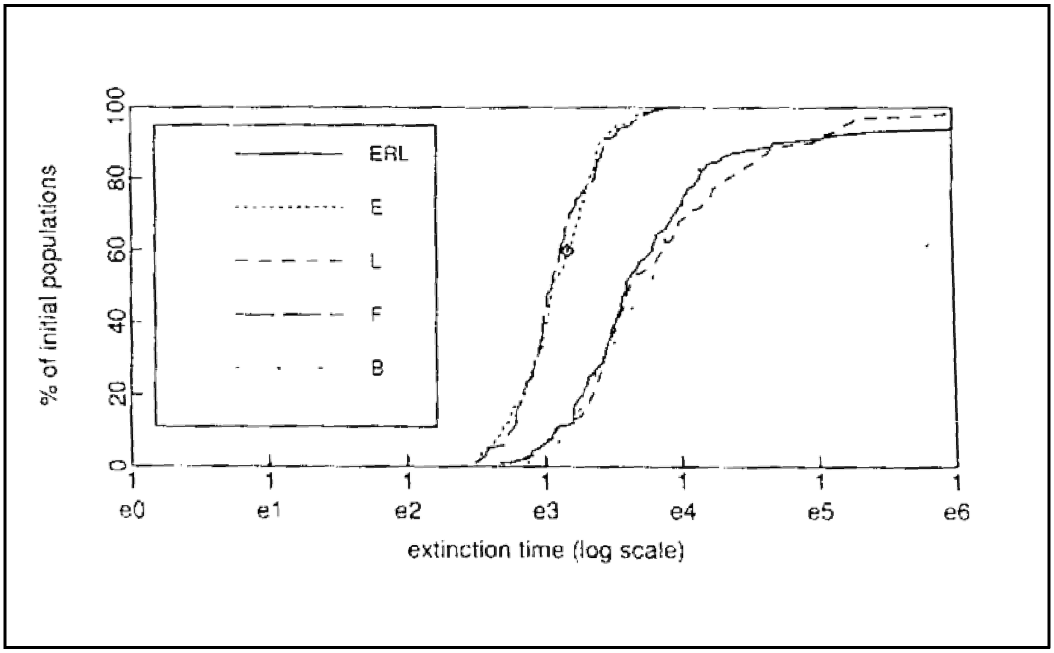

Figure 8: Performance of runs with random motion (B), fixed random action networks (F), learning alone (L), evolution alone (E), and the complete evolutionary reinforcement learning algorithm (ERL).

- Interactions between learning and evolution. The authors used functional constraints analysis, a traditional technique of evolutionary biology which studies the rate of change of different parts of the genome, to gain deeper knowledge about this model. This analysis centers on the notion that the more fixed parts of the genome are assumed to be important for survival during a given period. They observed genes associated with actions concerning food, evaluations concerning food, and genes coding for nothing (inserted for this sake):
    - Naturally, non-coding genes have the highest rate of mutation.
    - Evaluation genes mutate more, which implies that action genes are more functionally constrained. However, during the first stages of the simulation, evaluation genes are much more stable. Action genes become constrained later. This leads us to conclude that learning is essential in the first stages of the modeling experiment, but genetically encoded behavior for actions in the environment becomes more functionally constrained in latter stages: evidence for the *Baldwin effect*.

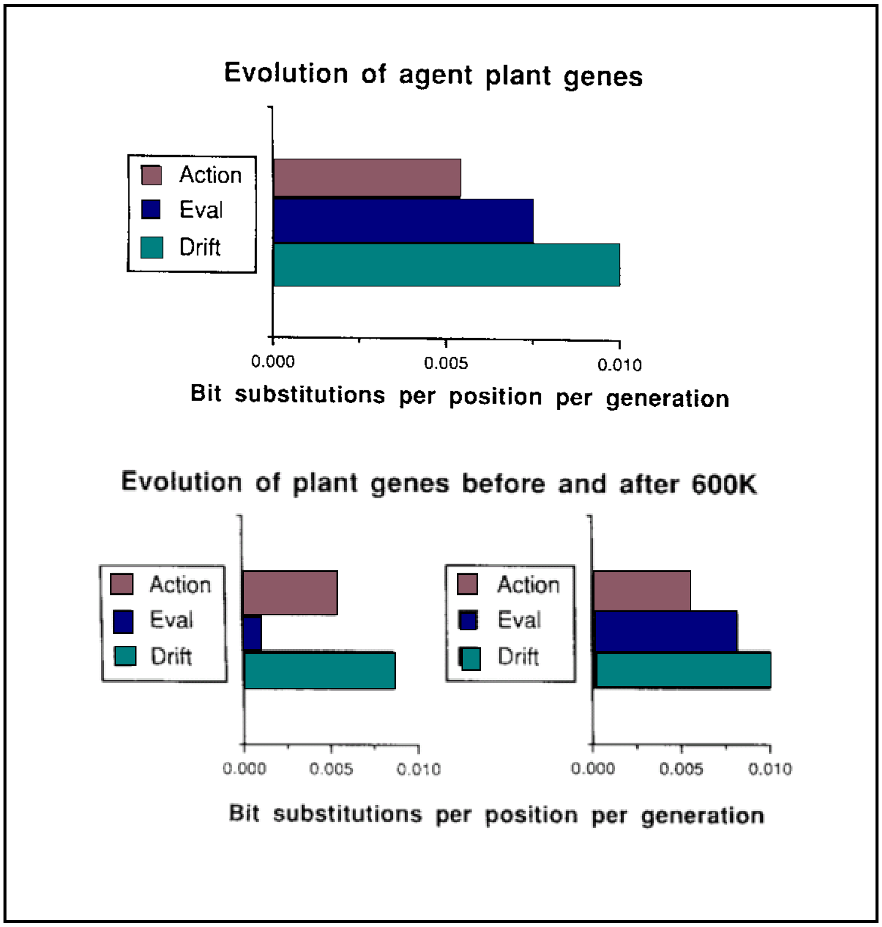

Figure 9: The top graph shows the average number of mutations (bit substitutions) per position per generation for an entire run and for plant agents. Action genes were the most functionally constrained. The bottom graphs separate the mutations before and after 600 generations. This shows that at first, evaluation genes were the most functionally constrained.

3.3 Evolution of Communication

Another class of models with interest to our problem area deals with the emergence and evolution of communication among agents with no explicitly programmed ability to communicate.

- EMERGENT PARTICLE COMPUTATION. Crutchfield and Mitchell [1995]. In this model, "agents" are simple boolean automata organized in a spatial one-dimensional lattice, or 1-D cellular automata (CA). The CA needs to solve a non-trivial task such as the density task: When the initial state of the whole lattice contains a majority of automata/agents with state=1 (0), then the whole lattice should converge to an all-1 (0) state. This is a non-trivial task because each agent has access solely to local information, namely the states of its neighbors, yet the lattice as a whole is expected to perform a global task. Clearly, the task can only be solved with some amount of information integration across the lattice. The experiments were set up in the following way:
    - Lattices of 149 (599, 999) boolean agent automata with boolean rules of radius 3 (7 cells in neighborhood).
    - A Genetic Algorithm (GA) was used to evolve rules to solve this task/environment.
    - A typical result from running the GA is the block-expansion rule (figure 10).

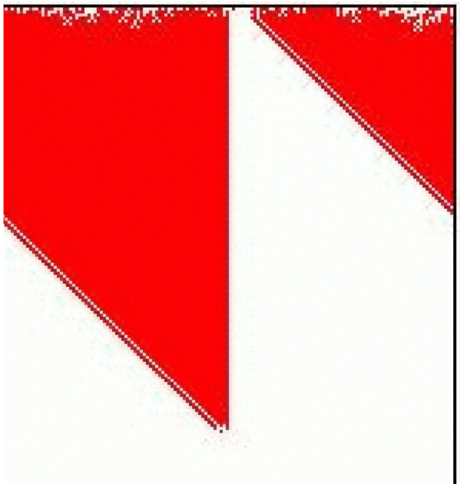

Figure 10:The block-expansion rule simply expands local neighborhoods with a large number of automata in the same state.

    - But occasionally a rule emerges that can solve this task with much better results. These rules create an intricate system of lattice communication. Groups of adjacent automata propagate certain patterns across the lattice, which as they interact with other such patterns "decide" on the appropriate solutions for the lattice as a whole. An intricate system of signaling patterns and its communication syntax has been identified, and can be said to establish the emergence of embedded-particle computation in evolved CA's [Crutchfield and Mitchell, 1995, Hordijk et al, 1996]. The emergent signals (or embedded particles) refer to the borders of the different patterns that develop in the space-time diagrams. If the areas inside these patterns are removed, their boundaries can be identified as a system of signals with a definite syntax, or emergent logic grammar. This syntax is based on a small number of signals, α, β, δ, γ, η, and μ, and a small number or rules such as: $\alpha + \delta \rightarrow \mu$, meaning that when signals α and δ collide, the μ signal results. See Figure 11.

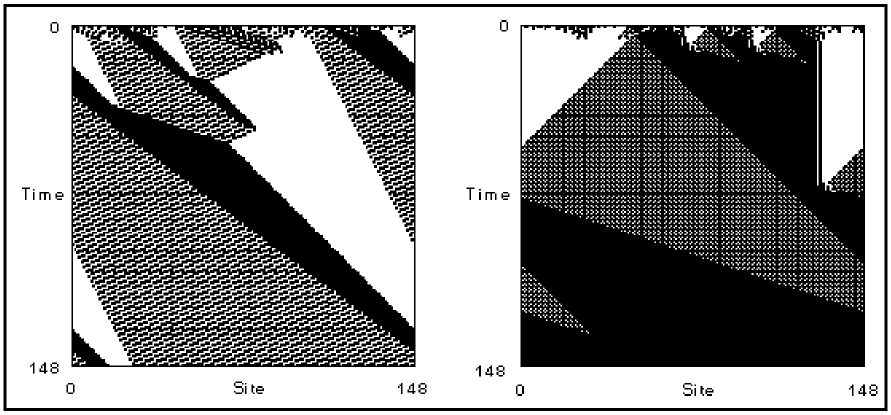

Figure 11: 2 examples of a particle computation rule applied to the density task on a lattice of 149 automata. The figure on the left depicts the case where the rule does not solve the initial configuration (it should converge to all white), while the figure on the right solves the task correctly for the particular initial configuration. The particles refer to the boundaries between the different patterns.

- The system of particle computation uses signals that are capable of integrating distant global information to solve the task. These CA rules rely on a system of signals used to communicate across the lattice and compute the answer to the task: an autonomous sign system that grants great selective advantage to the rules capable of developing it. The particle computation system truly introduces a qualitatively different way of solving the task: through the emergence of autonomous syntax, which allows certain rules to gain access to global lattice information [Rocha, 1998].
- Rocha[1998] has expanded these rules to solve more complicated tasks such as logical operations.

- EVOLUTION OF COMMUNICATION.
  - Werner and Dyer [1991], developed a agent-based computer experiment in the same line of thought as that of Ackley and Littman detailed above, to study the evolution of communication. The environment of these agents was set up so that blind male agents need to wander a 2-D environment in search of female agents with sight and the capacity to emit "sounds". Each agent has a distinct genome which is interpreted to produce a neural network that defines the agent's action behavior. This model also does not possess an explicit fitness function for the evolutionary algorithm. Rather, agents are given a certain amount of time to find a mate a reproduce. This produces the desirable asynchrony as agents do not reproduce all in the same time step. This results in overlapping generations with the possibility of inter-generational communication – important for the evolution and transmission of communication and knowledge. These experiments produced a number of very interesting "species" of male-female agents tuned to their own communication protocol or semantic closure. This model will undoubtedly also provide good design principles for our intended simulations.
  - *Evolution of Learning in the Cultural Process*. Hutchins and Hazlehurst [1991] produced an agent-based computer experiment to study how communication can and should emerge from the regularities of a given environment. The show that the evolution of communication is based adaptation of agents' internal structures to the structure of the environment, that is, coordination between internal and external structure. They embed their experiments in a framework which posits that culture is a *process* that permits the learning of previous generations to have direct effects on the learning of subsequent generations. Their experiments are also fundamentally semiotic, as they posit that the signals that enable communication of culture must be themselves somehow "physically" implemented in a environment. In other words, communication must itself be based on some structure that exists between agents and the environment in addition to agent internal structure and environmental structure. This third kind of structure is of an artifactual nature. This way, the agents of a cultural world have their internal structure shaped by (in coordination with) two kinds of structure in the environment: natural and artifactual. Their experiment is set up to produce agents that may learn a language (using a "physical code) able to mediate between environment regularities and artifactual descriptions of them. Their model sets up a simple environment (loosely based on ethnographic records of California Indians) where there is a relations between moon phase and tide state, which is relevant for agents feeding on shellfish. They use a cultural selection algorithm on agents that possess different neural networks for behavior and knowledge. They show that with the availability of artifactual structure, the learning of the relationship between moon phase and tide state is much more easily discovered and propagated in the population of agents. This model will undoubtedly also provide good design principles for our intended simulations.

- ○ Similar experiments based on culture and language were also performed by MacLennan [1991].

3.4 Agents with Shared Knowledge Structures

The last models of section 3.3 in their dealing with the evolution of communication, rely on the evolution of common structures among agents used to enable communication, e.g. artifactual structures. In this section we deal with models whose goal is to explicitly study the nature and consequences of such shared structure among agents. Being more explicit, these models emanate more from artificial intelligence, social science, and game theory, than from artificial life as the previous models did.

- THE COLLECTIVE CHOICE MODEL of Richards, McKay, and Richards [1998]. This model aims at the investigation of a fundamental problem of social choice models, particularly those used to model collective choice of political alternatives. Previously, it has been showed [e.g. Arrow, 1963, McKelvey, 1979] that the aggregation of elements into a collective choice when many agents and possible choices are involved typically leads to unstable, cyclic outcomes – for instance when we implement majority vote rules on automata. This is clearly at odds with reality where societies are capable of reaching stable political outcomes even in political systems with many available choices. The authors claim that such cyclic behavior exists because the previous system used to study collective choice do not make use of appropriate common structure among agents (this

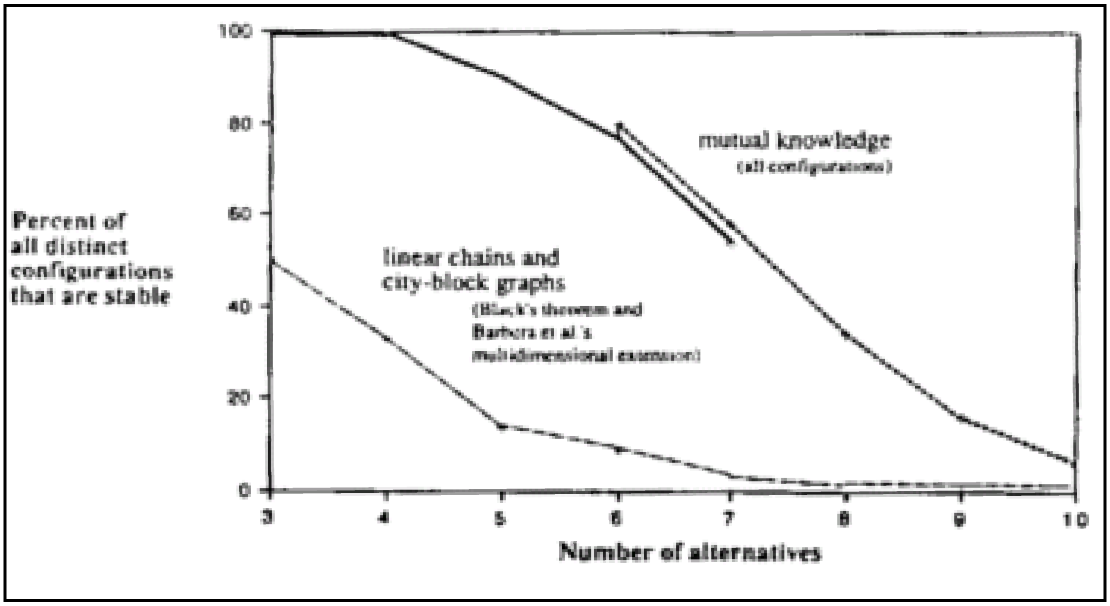

Figure 12:Percent of configurations that are stable under shared knowledge compared with linear constraints (with no constraints the number is zero according to Arrow's theorem).

claim is in line with the models of the previous section). Indeed, agents in previous models are assumed to make their own choices autonomously. The authors show mathematically and computationally that by establishing a shared knowledge structure, the probability of such cyclic behavior is reduced dramatically (figure 12).

○ The shared knowledge structure is implemented as a graph structure among alternatives. Such a graph implements the regularities of a particular choice landscape or environment. It is very reasonable to assume that alternatives are not all equally related (or un-related), but possess some structure. For instance, when deciding what kind of movie we want to watch, we all share some common knowledge: we know that a horror movie is more related to a violent action movie than to a romantic comedy. Likewise, we know that a Christian-democrat party is more related to a social democrat than to a communist party.

○ The shared knowledge structure implemented as a graph effectively establishes constraints on the choices which implement the regularities of a given choice system.

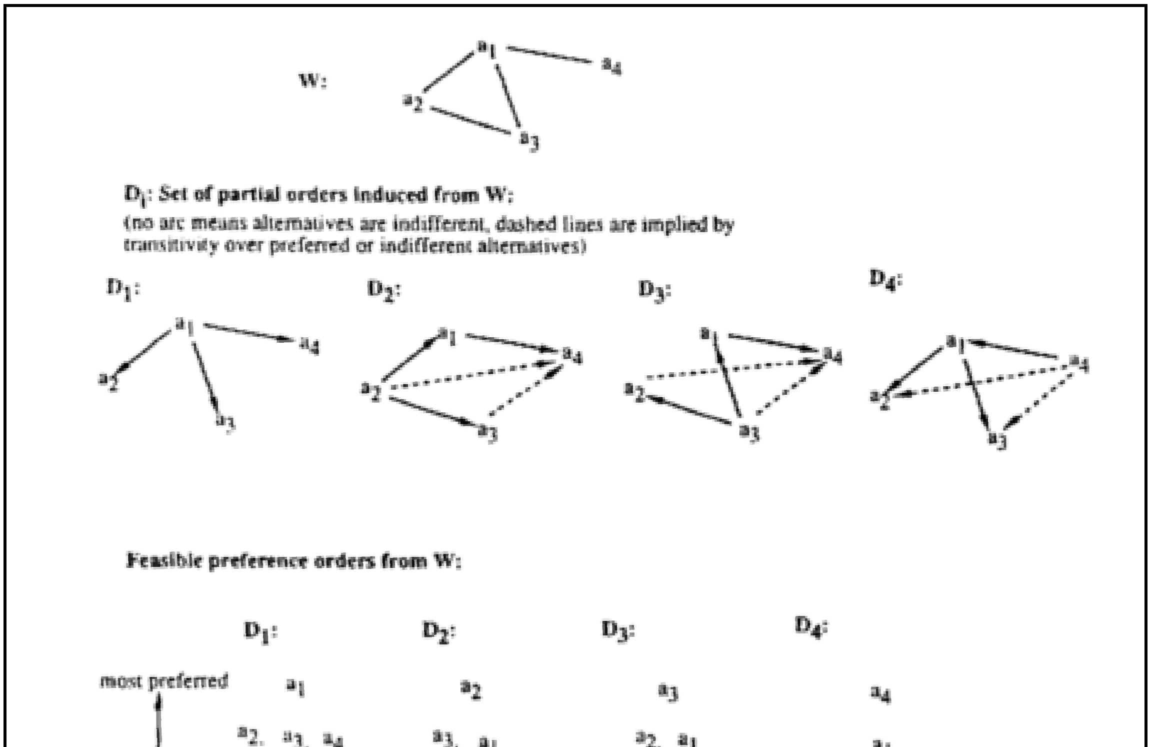
W:
a_1
a_2
a_3
a_4
D_i: Set of partial orders induced from W:
(no arc means alternatives are indifferent, dashed lines are implied by transitivity over preferred or indifferent alternatives)
D_1:
D_2:
D_3:
D_4:
Feasible preference orders from W:
most preferred
least preferred
D_1: a_1 / a_2, a_3, a_4
D_2: a_2 / a_3, a_1 / a_4
D_3: a_3 / a_2, a_1 / a_4
D_4: a_4 / a_1 / a_2, a_3

Figure 13: Example of a knowledge structure with 4 choices (above), and respective set of possible partial orders (below).

○ From the shared knowledge structure, different partial orders can be established to represent agents with different dispositions towards the shared knowledge structure. For instance, agents who prefer romantic comedies to horror movies, etc. See figure 13.

4. Developing Semiotic Agents

The architecture of semiotic agent models that we intend to design for this project will develop some ideas presented in some of the models described on the review above. Semiotic agents need to be based on a few fundamental characteristics of dynamical incoherency:

1. Asynchronous behavior.
2. Situated Communication.
3. Shared and cultural nature of language and knowledge.
4. Possession of evaluation capacity.
5. Localized, decoupled memory.

The asynchronous rules of the agent designs we are interested in are similar to the models of Ackley and Littman [1991] and the models of the evolution of communication. In these models agents do not reproduce or perform actions a constant time-steps, rather their actions follow discrete-event cues. This allows for inter-generational transmission of information, or more generally, the cohabitation of agents with different environmental experience. Furthermore, as Hutchins and Hazlhurst [1991], we assume that communication can only be based on the existence of environmental tokens and regularity which must follow the laws of the environment – situated communication (2).

The asynchrony and situated communication requirements establish localized constraints on communication and event-driven actions among agents. We wish to investigate the nature of these constraints. We expect these constraints to be similar to the shared knowledge constraints of the model of Richards et al [1998]. Notice that this model is based on synchronous updating. We postulate that asynchrony and situated communication will result in the emergence of a shared knowledge structure with the same characteristics of Richards' et al model. Situated communication and asynchrony will implement the realistic situation of having agents' choice be dependent on the choice their neighbors have pursued earlier as well as on the events in their neighborhood. This would be an important theoretical result for the study of choice models.

As for the agents internal architecture, we want to set them up with something akin to the agents of Ackley and Littman (which are similar to those of Werner and Dyer and Hutchins and Hazlhurst). In particular we want to separate between evaluation and action networks, so that we can model agents with different, independent beliefs about their present state and desirable goals (or the ability to evaluate their situation) (4). However, we want their evaluation networks to be constrained by shared knowledge structures either imposed on the set of alternatives or emergent from the asyncrony and situated communication.

Finally, eventually, we wish to endow agents with larger memory capacity so that we can model more realistically the kind of agents we require for this project.

5. REFERENCES

Ackley, D.H. and M. Littman [1991]."Interaction between learning and evolution." In: *Artificial Life II.* Langton et al (Eds). Addison-wesley, pp. 487-509.

Angeline, P.J. [1994]."An alternate interpretation of the iterated prisoner's dilemma and the evolution of non-mutual cooperation." In: *Artificial Life IV.* R. Brooks and P. Maes (Eds.). MIT Press, pp. 353-358.

Arrow, K.J. [1963]. *Social Choice and Individual Values.* .

Belew, R.K. [1990]."Evolution, learning, and culture: computational metaphors for adaptive algorithms." *Complex Systems.* Vol. 4, pp. 11-49.

Boyd, R., and J.P. Loberbaum [1987]."No pure strategy is evolutionarily stable in the repeated Prosiner's Dilemma game." *Journal of Theoretical Biology.* Vol. 136, pp. 47-56.

Crutchfield, J.P. and M. Mitchell [1995]."The evolution of emergent computation." *Proc. National Acadamy of Sciences, USA, Computer Sciences.* Vol. 92, pp. 10742-10746..

Emmeche, Claus [1994]. *The Garden in the Machine: The Emerging Science of Artificial Life.* Princeton University Press.

Farrell, J., andR. Ware [1989]."Evolutionary stability in the repeated Prisoner's Dilemma." *Theoretical Population Biology.* Vol. 36, pp. 161-166.

Hinton, G.E. and S.J. Nowlan [1987]."How learning can guide evolution." *Complex Systems.* Vol. 1, pp.495-502.

Holland, J.H. [1995]. *Hidden Order: How Adaptation Builds Complexity.* Addison-Wesley.

Hordijk, W., J.P. Crutchfield, and M. Mitchell [1996]."Embedded-particle computation in evolved cellular automata." In: *Proceedings of Physics and Computation 96.* . In Press. Available as a Santa Fe Institute preprint (96-09-073).

Hutchins, E. and B. Hazlehurst [1991]."Learning in the cultural process." In: *Artificial Life II.* C. Langton, C. Taylor, J.D. Farmer, and S. Rasmussen. Santa Fe Institute studies in the sciences of complexity series. Addison-Wesley, pp. 689-706..

Lindgren, K. [1991]."Evolutionary Phenomena in Simple Dynamics." In: *Artificial Life II.* Langton et al (Eds). Addison-wesley, pp. 295-312.

MacLennan, B. [1991]."Synthetic Ethology: An Approach to the study of communication." In: *Artificial Life II.* Langton et al (Eds.). Addison-wesley, pp. 631-558.

McKelvey, R.D. [1979]."General conditions for global intransitivities in formal voting models." *Econometrica.* Vol. 47, pp. 1085-112.

Mitchell, Melanie [1996]. *An Introduction to Genetic Algorithms.* MIT Press.

Oliphant, M. [1994]."Evolving cooperation in the non-iterated prisoner's dilemma: the importance of spatial organization." In: *Artificial Life IV.* R. Brooks and P. Maes (Eds.). MIT Press, pp. 349-352.

Pattee, Howard H. [1993]."The limitations of formal models of measurement, control, and cognition." *Applied mathematics and computation.* Vol. 56, pp. 111-130.

Richards, D., B.D. McKay, and W.A. Richards [1998]."Collective choice and mutual knowledge structures." *Advances in Complex Systems.* Vol. 1, pp. 221-236.

Rocha, Luis M. [1998]."Syntactic Autonomy." In: *Proceedings of the Joint Conference on the Science and Technology of Intelligent Systems (ISIC/CIRA/ISAS 98).* National Institute of Standards and Technology, Gaithersbutg, MD, September 1998. IEEE Press, pp. 706-711.

Sigmund, K. [1993]. *Games of Life: Explorations in ecology evolution and behavior.* Oxford University Press.

Stanley, E.A., D. Ashlock, and L. Tesfatsion [1994]."Iterated Prisoner's dilemma with choice and refusal of partners." In: *Artificial*

Life III. C.G. Langton (Ed.). Addison-Wesley, pp. 131-175.

Werner, G.M. and M. G. Dyer [1991]."Evolution of Communication in Artificial Organisms." In: *Artificial Life II.* Langton et al (Eds). Addison-wesley, pp. 659-687.